Galaxy Network Embedding: A Hierarchical Community Structure Preserving Approach

Lun Du[1]*, Zhicong Lu[1]*, Yun Wang[1]*, Guojie Song[1] †, Yiming Wang[1], Wei Chen[2]
[1] Peking University
[2] Microsoft Research
{dulun, phyluzhicong, wangyun94, gjsong, wangyiming17}@pku.edu.cn
weic@microsoft.com

Abstract

Network embedding is a method of learning a low-dimensional vector representation of network vertices under the condition of preserving different types of network properties. Previous studies mainly focus on preserving structural information of vertices at a particular scale, like neighbor information or community information, but cannot preserve the hierarchical community structure, which would enable the network to be easily analyzed at various scales. Inspired by the hierarchical structure of galaxies, we propose the Galaxy Network Embedding (GNE) model, which formulates an optimization problem with spherical constraints to describe the hierarchical community structure preserving network embedding. More specifically, we present an approach of embedding communities into a low dimensional spherical surface, the center of which represents the parent community they belong to. Our experiments reveal that the representations from GNE preserve the hierarchical community structure and show advantages in several applications such as vertex multi-class classification and network visualization. The source code of GNE is available online.

1 Introduction

Network embedding is a method of learning a low-dimensional representation of vertices in a complex network, under the condition of preserving different types of structural properties of the network. It enables effective applications of many general machine learning methods to network analyses, such as vertex classification, network clustering, network visualization and and social influence analysis [Bhagat *et al.*, 2011; He *et al.*, 2012; Song *et al.*, 2015].

Network embedding studies include structure-preserving methods [Grover and Leskovec, 2016; Wang *et al.*, 2017] and property-preserving methods [Hu *et al.*, 2017]. Our paper belongs to the former. In terms of structure-preserving methods, inspired by Skip-gram in word2vec [Mikolov *et al.*, 2013a], many methods consider the vertex context and represent a vertex with its nearby vertices [Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Tang *et al.*, 2015]. In addition to preserving the microscopic structure, the community structure, one important mesoscopic description of network structure, is incorporated into network embedding in MNMF, of which the learned embedding space can well reflect the organizational structures and functional components of networks [Wang *et al.*, 2017].

Essentially, these methods mainly focus on preserving neighborhood information, or community structure at a particular scale. However, the community structures in complex networks are often hierarchical [Clauset *et al.*, 2006], and many complex networks, such as social networks, air transportation networks, and metabolic networks, exhibit explicit hierarchical structures. Such a hierarchical structure greatly enriches the community structure and connects the community structure at different scales [Sales-Pardo *et al.*, 2007]. For instance, in a social network among college students, the hierarchical structure could include the university, the department within the university, and the enrollment year within the department, etc. Hierarchical network embedding aims at a succinct vector representation of the network that encodes the rich hierarchical structural information, which could make it much easier to analyze the network at different scales.

It is challenging to formulate the hierarchical community preserving network embedding. Not only the topological relationship between the nodes[1] in the same layer of the hierarchical tree needs to be considered, but the relationship between the nodes in different layers needs to be considered as well. In addition, efficiently implementing the embedding method is a challenge as well.

Conceptually, a hierarchical community network embedding should preserve two types of relationship information: One is the local information that represents the microscopic structure (pairwise nodes similarity) in the same community at a certain layer; the other is the hierarchical information. Horizontally, the nodes in the same community are more sim-

*These authors contributed equally to the work.

†Corresponding Author

[1] In this paper, we consistently use the term “vertex” to represent a node in the network, while using the term “node” to represent a node in the hierarchical tree.

ilar to each other. Vertically, a community in a lower level has a greater cohesion degree than a community in a higher level (e.g. in the solar system, the earth is closer to the moon, farther from the sun).

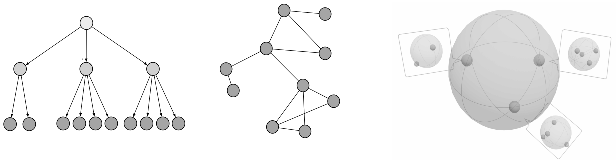
In this paper, inspired by the natural hierarchical structure of a galaxy with its stars, planets, and satellites, we propose the Galaxy Network Embedding (GNE) model. In particular, several properties of the galactic hierarchical structure influence our modeling and implementation, as we now explain. As most network embedding schemes, we formulate network embedding as an optimization task, with constraints on both the local community structure and the global hierarchical structure. For preserving pairwise similarity in local communities, we extend the LINE method [Tang *et al.*, 2015]. For global hierarchical structure constraints, we draw inspiration from the galactic hierarchy where celestial bodies belonging to the same subsystem have much shorter distance than (a) their distance to the same level celestial bodies in another subsystem, and (b) their distance to the center of a higher level system (e.g. satellites of one planet are much closer to one another than they are to satellites of a different planet, and than they are to the sun of their planet). We turn the first observation as a horizontal constraint and the second one as a vertical constraint and impose them for hierarchical network embedding. Moreover, when implementing the embedding, for the set of sibling nodes of the same parent node in the hierarchy, we embed them on a sphere with the parent at the center (Fig.1 (c)). This also resembles the galaxy structure, for example satellites of the same planet are at similar distance scale to the planet (thus on an approximate sphere centered at the planet), comparing to the distance scale between planets, and similar situation can be said for planets surrounding their sun. The spherical embedding of a subsystem in the hierarchy makes our embedding implementation efficient.

To summarize, we make the following contributions:

- We propose the Galaxy Network Embedding (GNE) model with horizontal and vertical constraints to preserve the hierarchical community structure at any scale.
- We present a novel spherical embedding method that leads to an efficient and effective implementation in the GNE model.
- We conduct extensive experiments on three real-world networks and four synthetic networks with explicit hierarchical structures, and the results demonstrate that our model can integrally preserve the hierarchical community structure and is significantly superior to other models on vertex classification and network visualization.

2 Related Work

Network Embedding. Network embedding maps the vertices or edges of a network into a low-dimensional vector space, which is beneficial to vertex classification and network visualization. It is well recognized that network representation has two goals: reconstructing the original network and supporting network inference. Manifold learning aims to reconstruct all the links, causing overfitting and limiting the network inference ability seriously [Tenenbaum *et al.*, 2000]. In order to support network inference, structure-preserving method and property-preserving method are proposed. In terms of structure-preserving methods, inspired by the word2vec in NLP [Mikolov *et al.*, 2013a], some methods consider the vertex context and represent a vertex with its nearby vertices [Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Tang *et al.*, 2015]. In addition to preserving the microscopic structure, the community structure, one important mesoscopic description of network structure, is incorporated into network embedding in MNMF [Wang *et al.*, 2017]. All the methods above mainly focus on preserving the pairwise proximity or community structure on a particular resolution, while the community structure at different scales are not considered.



(a) Tree

(b) Network

(c) Spherical Galaxy Model

Figure 1: Different display approaches of the network hierarchical structure

Hierarchical Network. Many complex networks in the real world have hierarchical community structures. [Newman, 2003] introduces the community structure of the network, and summarizes that complex networks have the small-world property and the scale-free property. [Song *et al.*, 2005] find that a large number of real networks have the self-similar property. Besides, the works on detection of hierarchial communities for networks with implicit hierarchical structure attract more attentions [Shen *et al.*, 2009]. Recently, there are few researches on network embedding taking into account the hierarchical structures of networks. [Nickel and Kiela, 2017] uses the hyperbolic space to study the hierarchical structure of the network. The learned representation vectors are in the hyperbolic space, which are difficult to be converted into the Euclidean vectors. However, most of machine learning algorithms require the input vectors to be in the Euclidean space. Thus, it is necessary to study hierarchical structure preserving network embedding in the Euclidean space.

3 Galaxy Network Embedding Model

In this section, we introduce our notation, and formulate hierarchical preserving network embedding to preserve the pairwise proximity between communities and the hierarchical structure properties. Inspired by the galaxy structure, we introduce the Galaxy Network Embedding.

3.1 Notation

An undirected network G is denoted as (V, E), where V is the set of vertices and E is the set of edges. The hierarchical clustering tree of G is denoted as T with a depth of L. C denotes the node set and c_i^l denotes the i-th node at the l-th level of T. $Ch(c)$ denotes the child node set of the node c

and $pa(c)$ denotes the parent node of c. Meanwhile, c_i^l also represents the i-th community while recursively dividing G at the l-th level. For the l-th level of T,

$$\forall c_i^l, c_j^l \in C^l \quad c_i^l \cap c_j^l = \varnothing,$$
$$\bigcup_i c_i^l = V.$$

Especially, c_1^1 is the root node of T and also the node set of G (i.e. $c_1^1 = V$) and $c_i^L = \{v_i\}$, where v_i is the i-th vertex of the G.

In order to preserve the hierarchical structure property of the network explicitly, we embed not only the vertices but also the communities of all layers. Due to $c_i^L = \{v_i\}$, for brevity, we use the community representation instead of the vertex representation in the model.

We denote $\Phi(c) \in \mathbb{R}^m$ as the representation of the community c, where $m \ll |C|$. The problem of the hierarchical preserving network embedding is to learn the low-deminsional representations Φ of communities that can preserve the pairwise proximity between communities and the hierarchical structure properties.

3.2 Hierarchical Preserving Network Embedding

In our scenario, the community representation can preserve two kinds of properties of G. One is the local information, i.e. the pairwise proximity between sub-communities in the same community at a certain layer. The other is the hierarchical structure property, i.e. horizontal relationship and vertical relationship.

Pairwise Proximity Preservation

In order to preserve the local information between communities, we first define the community proximity $S_{i,j}^k$ extended from the definition of the common neighbor similarity [Libennowell and Kleinberg, 2007]:

$$S_{i,j}^l = \frac{1}{|c_i^l||c_j^l|} \sum_{u \in c_i^l} \sum_{v \in c_j^l} \frac{A_u^T A_v}{\sqrt{||A_u||_1 ||A_v||_1}}, \tag{1}$$

which means the average common neighbor similarity between the community c_i^l and c_j^l, where A is the adjacency matrix of G and A_u is the u-th column of the A.

We extend LINE with Second-order Proximity [Tang *et al.*, 2015] to define a local objective for the communities deriving from the same parent node c_k^{l-1}, namely,

$$\min_{\Phi,\Phi'} O_k^{(l-1)} = - \sum_{c_i^l, c_j^l \in Ch(c_k^{l-1})} S_{i,j}^l \log P(\Phi'(c_j^l) \,|\, \Phi(c_i^l)), \tag{2}$$

which minimizes the KL-divergence between the distribution of community proximity from the ground truth and from the learned representation respectively. $P(\Phi'(c_j^l) \,|\, \Phi(c_i^l))$ is modeled by the softmax function, i.e.

$$P(\Phi'(c_j^l) \,|\, \Phi(c_i^l)) = \frac{\exp(\Phi'(c_j^l) \cdot \Phi(c_i^l))}{\sum_{c_t^l \in Ch(c_k^{l-1})} \exp(\Phi'(c_t^l) \cdot \Phi(c_i^l))},$$

where $\Phi(c)$ is the representation of c and $\Phi'(c)$ is an auxiliary vector defined in LINE. Considering the computational efficiency, we approximate the softmax function using negative sampling [Mikolov *et al.*, 2013b].

Hierarchical Structure Preservation

Besides the pairwise proximity, the community representations should preserve the hierarchical structure property. In detail, two relationships should be preserved in the hierarchical tree:

Horizontal relationship: nodes belonging to the same community should be closer to each other than those belonging to different communities in a certain layer, which can be illustrated with the following equation in Euclidean space: For all $l = 1, \ldots, L$, for all $c_u^l, c_v^l \in C^l$ with $pa(c_u^l) = pa(c_v^l)$, for all $c_w^l \in C^l$ with $pa(c_u^l) \neq pa(c_w^l)$, we have

$$\|\Phi(c_u^l) - \Phi(c_v^l)\| < \|\Phi(c_u^l) - \Phi(c_w^l)\|, \tag{3}$$

where $\|\cdot\|$ means 2-norm.

Vertical relationship: the cohesion degree of shallower layer communities should be smaller than that of deeper layer ones in T. Naturally, the cohesion degree of a community can be represented by the average representations distance between the sub-communities in it. Thus, the parent-child relationship can be described with the following equation: For all $l = 1, \ldots, L-1$, for all $c_k^{l-1} \in C^{l-1}$, for all $c_i^{l+1} \in Ch(c_j^l)$ and all $c_j^l \in Ch(c_k^{l-1})$, we have

$$\|\Phi(c_i^{l+1}) - \Phi(c_j^l))\| < \|\Phi(c_j^l) - \Phi(c_k^{l-1})\|. \tag{4}$$

Considering the pairwise proximity preservation (Eq.(2)) and hierarchical structure preservation (horizontal Eq.(3), vertical Eq.(4)), we introduce Galaxy Network Embedding to learn the representations of nodes in hierarchical tree.

3.3 Galaxy Network Embedding

As discussed in the introduction, we resemble the galactic hierarchical structure to use sphere embedding for sibling nodes in the hierarchical structure. This leads to the following GNE optimization formulation:

$$\begin{aligned} &\min_{\Phi,\Phi'} \quad O_k^{(l-1)}(\Phi, \Phi') \\ &s.t. \quad \forall c_i^l \in Ch(c_k^{l-1}), \quad \|\Phi(c_i^l) - \Phi(c_k^{l-1})\|_2 = r_k^{l-1}. \end{aligned} \tag{5}$$

The r_k^{l-1} in the above formulation denotes the sphere radius of the community c_k^{l-1}, to reflect the spherical embedding

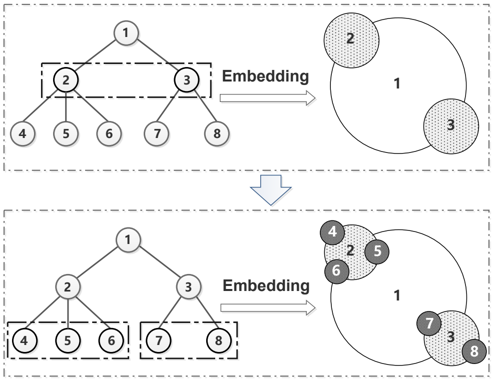


Figure 2: Structure of GNE

where all children nodes of c_k^{l-1} are embedded at a sphere of radius r_k^{l-1} centered at c_k^{l-1}. The radius r_i^l of the community c_i^l is obtained by the following strategy:

$$r_i^l = \eta \cdot d_k^{l-1}, \tag{6}$$

where,

$$\begin{aligned} d_k^{l-1} &= \min_{c_i^l, c_j^l \in Ch(c_k^{l-1}), i \neq j} Dist\left(\Phi(c_i^l), \Phi(c_j^l)\right), \\ Dist(x, y) &= \|x - y\|, \end{aligned} \tag{7}$$

and $pa(c_i^l) = c_k^{l-1}$, η is an arbitrary positive real number less than $\frac{1}{6}$. Especially, d_k^{l-1} is determined when calculating r_i^l.

In this way, we can optimize the objective Eq.(5) recursively from top to bottom in hierarchy to preserve the pairwise proximity and the hierarchical structure. The whole embedding procedure is illustrated in Fig.2. Obviously, the pairwise proximity can be preserved with the GNE objective Eq.(5). Next, we introduce Lemma.1 to prove the hierarchical structure can be preserved with Eq.(5).

Lemma 1. The community representations learned from recursively optimizing the Objective Eq.(5) with the strategy Eq.(6) preserve the constraints Eq.(3) and Eq.(4).

Proof. According to Eq.(5) and Eq.(6): for any $c_i^l \in C$,

$$r_i^l < \frac{1}{6} d_k^{l-1} \leq \frac{1}{6} \cdot 2 r_k^{l-1} = \frac{1}{3} r_k^{l-1}, \tag{8}$$

where $pa(c_i^l) = c_k^{l-1}$.

For any nodes c_u^l, c_v^l and c_w^l at layer l, where $pa(c_u^l) = pa(c_v^l)$ and $pa(c_u^l) \neq pa(c_w^l)$. $c_k^{\tau-1}$ is the common ancestor of c_u^l and c_w^l. Besides, the $c_k^{\tau-1}$'s children c_i^τ and c_j^τ are ancestors of the c_u^l and c_w^l, respectively. Thus, according to the constraints in Eq.(5) and Eq.(8):

$$\begin{aligned} Dist(\Phi(c_u^l), \Phi(c_i^\tau)) &\leq \sum_{c_e^\gamma \in P(c_i^\tau, c_u^l)} r_e^\gamma < r_i^\tau \sum_{\gamma=\tau}^{l-1} (\frac{1}{3})^{\gamma-\tau} \\ &< r_i^\tau \sum_{\gamma=\tau}^{\infty} (\frac{1}{3})^{\gamma-\tau} = \frac{3}{2} r_i^\tau < \frac{3}{2}\left(\frac{1}{6} d_k^{\tau-1}\right) = \frac{1}{4} d_k^{\tau-1}, \end{aligned}$$

where, $P(c_i^\tau, c_u^l)$ is a set of nodes in all paths from c_i^τ to c_u^l in the hierarchical tree, $c_i^\tau \in P(c_i^\tau, c_u^l)$ and $c_u^l \notin P(c_i^\tau, c_u^l)$

$$Dist(\Phi(c_w^l), \Phi(c_j^\tau)) < \frac{1}{4} d_k^{\tau-1}.$$

According to Eq.(7):

$$Dist(\Phi(c_i^\tau), \Phi(c_j^\tau)) \geq d_k^{\tau-1}.$$

Thus,

$$\begin{aligned} & Dist(\Phi(c_w^l), \Phi(c_u^l)) \geq Dist(\Phi(c_i^\tau), \Phi(c_j^\tau)) \\ & \quad - Dist(\Phi(c_i^\tau), \Phi(c_u^l)) - Dist(\Phi(c_j^\tau), \Phi(c_w^l)) > \frac{1}{2} d_k^{\tau-1}. \end{aligned}$$

Further, we can obtain the following relationship according to Eq.(6) and Eq.(8):

$$Dist(\Phi(c_u^l), \Phi(c_v^l)) \leq 2 r_{pa(c_u^l)}^l < \frac{1}{3} d_k^{\tau-1}.$$

Finally, we can obtain

$$Dist(\Phi(c_w^l), \Phi(c_u^l)) > Dist(\Phi(c_u^l), \Phi(c_v^l)).$$

Thus, we prove the community representations learned from GNE can preserve the horizontal relationship in hierarchy. Similarly, we can prove that the vertical relationship can be also preserved and we don't repeat it in the paper. □

4 Learning Procedure of GNE

In this section, we introduce the learning algorithm of GNE, including how to solve the optimization problem Eq.(5) layer by layer and the pseudocode of GNE. Compared with the optimization objective of the classic neural embedding method, we just add an extra spherical constraint in Eq.(5). The neural embedding method can be directly optimized using SGD, which is efficient and the effect of which has been verified. Considering these advantages, we transform Eq.(5) into a two-step optimization procedure. First, we use the Adam optimizer to optimize Eq.(2) without constraint based on neural networks, and obtain the intermediate representation $\Omega(c_i^l)$ of communities that preserves the pairwise relationship of communities. Second, we map $\Omega(c_i^l)$ to a spherical surface to get the final representation $\Phi(c_i^l)$ of communities. Next, we introduce the spherical projection method.

4.1 Spherical Projection

To preserve the relative distances in spherical projection, we formally define the following optimization objective:

$$\begin{aligned} \min_{\Phi} J_k^{(l-1)} = \left\| \frac{D}{\|D\|_F} - \frac{B}{\|B\|_F} \right\|_F + \mu \exp(-\gamma \|B\|_F) \\ s.t. \|\Phi(c_i^l) - \Phi(c_k^{l-1})\| = r_k^{l-1}, \end{aligned} \tag{9}$$

where,

$$\begin{gathered} c_i^l, c_j^l \in Ch(c_k^{l-1}), \\ D_{ij} = \|\Omega(c_i^l) - \Omega(c_j^l)\| \quad and \quad B_{ij} = \|\Phi(c_i^l) - \Phi(c_j^l)\|. \end{gathered}$$

The first term is to preserve the relative distances, and the second term is a penalty term to make the distances between each pair of points after projection are as large as possible. μ and γ are hyper parameters. As the overall optimization procedure is top-down, $\Phi(c_k^{l-1})$, r_k^{l-1} and D are all constants.

Spherical Projection Optimization Procedure

The constraints of Eq.(9) are for arbitrary spherical surfaces. We can perform scaling and translation transformations on the coordinate system to turn them into the unit sphere constraints, namely the following conversion :

$$\Psi(c_i^l) = \frac{\Phi(c_i^l) - \Phi(c_k^{l-1})}{r_k^{l-1}} \quad and \quad \|\Psi(c_i^l)\| = 1. \tag{10}$$

Combining with Eq.(10), the Euclidean distance between $\Psi(c_i^l)$ and $\Psi(c_j^l)$ can be expanded as follows:

$$\begin{aligned} B'_{ij} &= \|\Psi(c_i^l) - \Psi(c_j^l)\| \\ &= \|\Psi(c_i^l)\|^2 + \|\Psi(c_j^l)\|^2 - 2\Psi(c_i^l)^T \Psi(c_j^l) \\ &= 2 - 2\cos(\Psi(c_i^l), \Psi(c_j^l)). \end{aligned} \tag{11}$$

It can be seen that B'_{ij} is not related to the length of vector $\Psi(c_i^l)$ and $\Psi(c_j^l)$, but only the cosine distance between them. Therefore, the vector length constraint in the objective can be removed. The final optimization objective on the unit sphere is:

$$\min_{\Psi} H_k^{(l-1)} = \left\| \frac{D}{\|D\|_F} - \frac{B'}{\|B'\|_F} \right\|_F + \mu \exp(-\gamma \|B'\|_F). \tag{12}$$

We use $\Psi_*(c_i^l)$ to denote the minimum point after optimization. After normalization, translation, and scaling, the final results are obtained:

$$\Phi(c_i^l) = \frac{\Psi_*(c_i^l)}{\|\Psi_*(c_i^l)\|} \times r_k^{l-1} + \Phi(c_k^{l-1}). \tag{13}$$

Obviously, the unconstrained objective has a lower bound that can be found using Gradient Descent algorithm.

4.2 The GNE Algorithm

The pseudocode for GNE, is given in Algorithm 1. The whole embedding process is executed from top to down. For all children nodes of a certain node c_i^l, i.e. the nodes in $Ch(c_i^l)$, the learned representation $\Psi(Ch(c_i^l))$ on the unit sphere can be obtain after two-step optimization procedure with Adam, the optimization Eq.(2) is used first and the optimization Eq.(12) follows. The final representation $\Phi(c_j^{l+1})$ can be obtained by the transformation in Eq.(13). Besides, the corresponding r_j^{l+1} can be calculated with the minimum distance $d_{j,*}$. The constant η less than $\frac{1}{6}$ (see Lemma 1), so $\eta = \frac{1}{7}$ in our experiments.

Algorithm 1 The GNE algorithm

```
function LEARNFEATURES(Network $G$, Hierarchical Clustering Tree $T$)
    Initialize $\Phi(c_1^1) = 0$ and $\mathbf{r_1^1} = \mathbf{1}$
    $\mathbf{\Phi}$ = RecursiveOptimization ($c_1^1$, $G$, $T$, $\mathbf{r}$, $\mathbf{\Phi}$)
return $\mathbf{\Phi}$

function RECURSIVEOPTIMIZATION(Current node $c_i^l$, Network $G$, Hierachical Clustering Tree $T$, Radius Set $\mathbf{r}$, Reresentation Set $\mathbf{\Phi}$)
    if $l = L$ then return $\mathbf{\Phi}$
    $\Omega(Ch(c_i^l))$ = minimize $O_i^{(l)}$ with Adam (see Eq.(2))
    $\Psi(Ch(c_i^l))$ = minimize $H_i^{(l)}$ with Adam (see Eq.(12))
    for all $c_j^{l+1} \in Ch(c_i^l)$ do
        $\Phi(c_j^{l+1}) = \Psi(c_j^{l+1})/\|\Psi(c_j^{l+1})\| \times \mathbf{r_i^l} + \mathbf{\Phi(c_i^1)}$
        $\mathbf{r_j^{l+1}} = \eta \cdot \mathbf{d_{j,*}}$
    for all $c_j^{l+1} \in Ch(c_i^l)$ do
        $\mathbf{\Phi}$ = RecursiveOptimization ($c_j^{l+1}$, $G$, $T$, $\mathbf{r}$, $\mathbf{\Phi}$)
    return $\mathbf{\Phi}$
```

Time Complexity Analysis

For brevity of description, we assume that the hierarchical clustering tree of network $G = (V, E)$ is a k-ary tree T with depth $log_K|V|$. The number of nodes N in T is $\frac{K^{log_K|V|+1}-1}{K-1}$. Spherical embedding should be done on each tree node layer by layer. Especially, the embedding procedures on the same layer tree nodes can be parallelized. For spherical embedding, the complexity of computing $\Omega(Ch(c_i^l))$ is $O(\mathcal{E}KQ)$. More precisely, $\mathcal{E}$ is the number of training epoch, K is the size of community to be embedded and Q is the neural network computing complexity of skip-gram model, i.e. $Q = O(C(\mathcal{D} + \mathcal{D}M))$, where C is the skip window size, $\mathcal{D}$ is the embedding size and M is the negative sampling size; the complexity of computing $\Psi(Ch(c_i^l))$ is $O(\mathcal{E}K^2\mathcal{D})$. Thus, the spherical embedding complexity is $O(\mathcal{E}KQ + \mathcal{E}K^2\mathcal{D})$. The optimization algorithm is implemented on the Tensorflow platform, which can be accelerated with GPU.

5 Experiments

In this section, we present our experimental evaluation of the GNE method on both synthetic and real networks.

5.1 Experiment Setup

Data Sets We employ the following three real datasets in the Facebook social networks dataset which comprises 100 colleges and universities in US [Traud *et al.*, 2012]. We choose the social networks in Hamilton University (2314 nodes, 96394 edges), Amherst College (2235nodes, 90954edges) and Georgetown Universit (9414 nodes, 425639 edges). Moreover, in order to evaluate the performance of hierarchical community structure preservation, four Hierarchical Random Graphs (HRG) with explicit hierarchical community structure are generated by [Clauset *et al.*, 2008], i.e. Syn_with_125nodes (125 nodes, 406 edges), Syn_with_1800nodes (1800nodes, 739637 edges), Syn_with_2560nodes (2560 nodes, 1460147 edges) and Syn_with_3750nodes (3750 nodes, 3066250 edges).

Relevant Algorithms We compare the GNE against six network embedding algorithms: SpectralClustering [Tang and Liu, 2011], DeepWalk [Perozzi *et al.*, 2014], Node2Vec [Grover and Leskovec, 2016], LINE [Tang *et al.*, 2015], GraRep [Cao *et al.*, 2015] and MNMF [Wang *et al.*, 2017].

Parameters Settings The hyper-parameters of GNE is $\boldsymbol{\theta}$, i.e. $\boldsymbol{\theta} = (\mu, \gamma)$. GNE is not very sensitive to the hyper-parameters, the settings of which can achieve ideal results with grid search on a small range. In our experiments, the embedding size m of all models is 64. Besides, the parameter setting of comparison models follow the recommended settings in relevant code packages.

5.2 Hierarchical Community Detection

In this section, we verify the ability of hierarchical community preservation of our model GNE. We consider synthetic data sets in this experiment, including three different structures of HRG but with the same number of layers (see Fig.3). Jaccard's coefficient is applied as an external index for evaluating the performance of community preservation at each hierarchy of networks.

Fig.3 shows that the content of the hierarchical community structure can be integrally preserved with our model, no matter how many communities are there. However, MNMF only

Model	Amherst					Hamilton					Georgetown				
	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%	10%	30%	50%	70%	90%
GNE	**93.57**	**93.31**	**93.33**	**93.18**	**92.89**	**94.83**	**94.53**	**94.11**	**94.17**	**93.86**	53.22	**53.80**	53.55	**52.20**	**51.88**
SpectralClustering	72.89	73.49	73.94	74.32	72.82	78.16	77.60	77.21	77.59	74.92	49.26	50.87	50.79	50.60	48.53
DeepWalk	90.62	91.65	91.32	91.13	90.41	92.89	92.33	92.52	92.18	91.55	54.07	53.79	53.35	51.69	50.92
Node2Vec	91.29	91.24	91.04	90.44	90.02	92.09	91.03	91.18	90.06	89.56	52.86	53.73	53.16	52.70	51.28
LINE	90.76	91.82	91.48	91.09	89.42	92.33	92.72	92.52	92.62	91.73	54.64	53.45	53.81	52.71	51.28
GraRep	92.13	92.25	91.78	91.56	91.48	93.67	93.04	92.30	92.40	91.00	**54.80**	53.24	**53.95**	51.87	51.74
MNMF	89.82	89.06	88.04	86.43	78.44	91.42	90.32	89.12	87.02	81.19	53.43	52.63	52.10	51.52	50.35

Table 1: The multi-label classification results on different percentages of test datasets

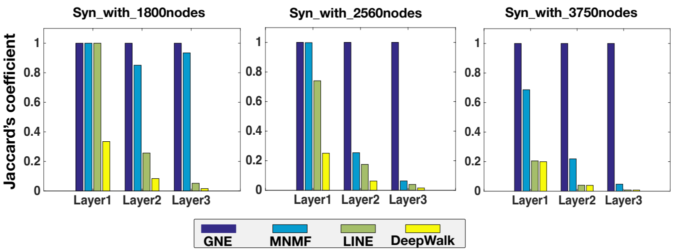
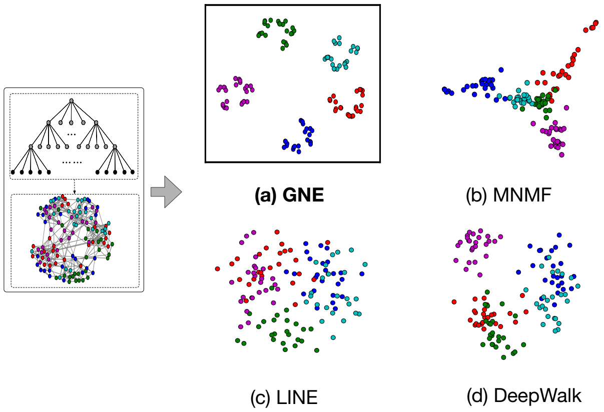


Figure 3: The comparison of hierarchical community preservation on different models. Three different structures of HRG with the same number of layers are used.



(a) GNE

(b) MNMF

(c) LINE

(d) DeepWalk

Figure 4: The visualization of vertex representations on different models

preserves community information at some layers but not all, and its performance is inferior to GNE, while other models perform much worse and could not handle multi-layer and complex community structure well.

5.3 Network Visualization

Network visualization is an important application of network embedding, and it maps a network into the two-dimensional space. We visualize a synthetic network with 125 nodes, 406 edges and 5 communities. Fig.4 presents the visualization experiments. We firstly generate a self-similar network of which nodes are derived from the leaves of five-ary tree with four layers and edges are derived from the sampling of connected paths between leaves in the tree. Additionally, the vertices in the network are classified into different communities with GirvanNewman algorithm [Girvan and Newman, 2002]. We compare our method against other models. For other models, we layout the network into low-dimensional space, and then further map the low-dimensional vectors of the vertices to a 2-D space with t-SNE package [Maaten and Hinton, 2008]. For our model, we can directly embed network into 2-D vector space based on our spherical embedding we proposed.

It can be seen from Fig.4 that our model GNE embeds vertices on the different-scaled spherical surface hierarchically. It is evident that the vertex representations of GNE are consistent with modularity property at each hierarchy, i.e. higher intra-cluster similarity but lower inter-cluster similarity. That is to say, GNE integrally preserves the hierarchical community structure. Additionally, GNE has an outstanding performance on clustering vertices compared with others.

5.4 Vertex Classification

In order to verify the effectiveness of GNE on vertex classification, three real social networks of the Facebook datasets with the four-layer hierarchical tree (including root and leaves) are used. The two intermediate layers are divided by the enrollment year and major, respectively. For MNMF, we use enrollment year as an indicator for community division. The learned representations are used to classify the vertices into a set of labels. The classifier we used is Logistic Regression with sklearn package, and the evaluation metric is Accuracy. Different percentage of nodes are sampled randomly for evaluation, and the rest are for training. The results are averaged over 10 different runs.

Table 1 shows that GNE almost always performs the best among all models on different percentage of test data size. Our model is also robust across different percentage of test datasets.

6 Conclusion

In this paper, we propose Galaxy Network Embedding (GNE) for network embedding to preserve the hierarchical community structure of a network. Specifically, we introduce an optimization problem with constraints and transform it into an unconstrained optimization problem more easily to be solved. Moreover, we propose a spherical embedding method to maintain the hierarchical community structure from top to bottom. Empirically, we verify GNE in a variety of network datasets and applications. The extensive experimental results on vertex clustering and classification, as well as network visualization, demonstrate the advantages of GNE, especially on the networks with hierarchical community structures.

Acknowledgments

We are thankful to Yizhou Sun for her helpful suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 61572041 and No. 61433014)

References

[Bhagat *et al.*, 2011] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. *arXiv: Social and Information Networks*, pages 115–148, 2011.

[Cao *et al.*, 2015] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *ACM International on Conference on Information and Knowledge Management*, pages 891–900, 2015.

[Clauset *et al.*, 2006] Aaron Clauset, Cristopher Moore, and M E J Newman. Structural inference of hierarchies in networks. *international conference on machine learning*, pages 1–13, 2006.

[Clauset *et al.*, 2008] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *arXiv preprint arXiv:0811.0484*, 2008.

[Girvan and Newman, 2002] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Knowledge Discovery and Data mining*, pages 855–864, 2016.

[He *et al.*, 2012] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. pages 463–474, 2012.

[Hu *et al.*, 2017] Xia Hu, Xia Hu, and Xia Hu. Label informed attributed network embedding. In *Tenth ACM International Conference on Web Search and Data Mining*, pages 731–739, 2017.

[Libennowell and Kleinberg, 2007] David Libennowell and Jon M Kleinberg. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[Maaten and Hinton, 2008] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2605):2579–2605, 2008.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Neural Information Processing Systems*, pages 3111–3119, 2013.

[Newman, 2003] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[Nickel and Kiela, 2017] M. Nickel and D. Kiela. Poincar\'e Embeddings for Learning Hierarchical Representations. *ArXiv e-prints*, May 2017.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Alrfou, and Steven Skiena. Deepwalk: online learning of social representations. *Knowledge Discovery and Data mining*, pages 701–710, 2014.

[Sales-Pardo *et al.*, 2007] Marta Sales-Pardo, Roger Guimer, Andr A. Moreira, and Lus A. Nunes Amaral. Extracting the hierarchical organization of complex systems. *arXiv preprint arXiv:0705.1679*, 104(39):15224–15229, 2007.

[Shen *et al.*, 2009] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A Statistical Mechanics & Its Applications*, 388(8):1706–1712, 2009.

[Song *et al.*, 2005] Chaoming Song, Shlomo Havlin, and Hernan A Makse. Self-similarity of complex networks. *arXiv preprint cond-mat/0503078*, 2005.

[Song *et al.*, 2015] Guojie Song, Xiabing Zhou, Yu Wang, and Kunqing Xie. Influence maximization on large-scale mobile social network: A divide-and-conquer method. *IEEE Transactions on Parallel and Distributed Systems*, 26(5):1379–1392, 2015.

[Tang and Liu, 2011] Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *International Conference on World Wide Web*, pages 1067–1077, 2015.

[Tenenbaum *et al.*, 2000] J. B. Tenenbaum, Silva V De, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2000.

[Traud *et al.*, 2012] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *Social Science Electronic Publishing*, 391(16):4165–4180, 2012.

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Association for the Advancement of Artificial Intelligence Conference*, 2017.